

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-350735

(43)Date of publication of application : 21.12.2001

(51)Int.Cl.

G06F 15/177

G06F 11/30

G06F 15/16

(21)Application number : 2000-172975

(71)Applicant : BOSCH AUTOMOTIVE SYSTEMS CORP

(22)Date of filing : 09.06.2000

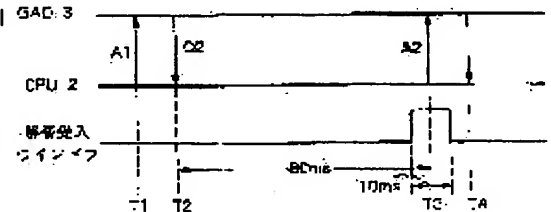
(72)Inventor : ONISHI AKIHIRO

(54) METHOD FOR MUTUAL MONITORING AMONG PLURAL DATA PROCESSORS

(57)Abstract:

PROBLEM TO BE SOLVED: To speedily detect even slight trouble by mutually monitoring operation between data processors.

SOLUTION: A question number Q2 selected at random is sent from a CPU 2 to a GAD 3, and the GAD 3 performs arithmetic determined corresponding to the question number Q2 and sends the arithmetic result A2 to the CPU 2. The CPU 2 increases/decreases the value of an error counter 31 corresponding to whether the arithmetic result A2 is sent within a prescribed period opening an answer accepting window or not and whether the contents of the arithmetic result A2 show a right answer or not and changes control operation corresponding to the result.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

THIS PAGE BLANK

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2001-350735
(P2001-350735A)

(43) 公開日 平成13年12月21日 (2001. 12. 21)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
G 0 6 F 15/177	6 7 8	G 0 6 F 15/177	6 7 8 A 5 B 0 4 2
11/30		11/30	F 5 B 0 4 5
15/16	6 2 0	15/16	6 2 0 C

審査請求 未請求 請求項の数 5 O L (全 5 頁)

(21) 出願番号 特願2000-172975(P2000-172975)

(22) 出願日 平成12年6月9日 (2000. 6. 9)

(71) 出願人 000003333

株式会社ボッシュオートモーティブシステム

東京都渋谷区渋谷3丁目6番7号

(72) 発明者 大西 章浩

埼玉県東松山市箭弓町3丁目13番26号 株式会社ゼクセル東松山工場内

(74) 代理人 100077540

弁理士 高野 昌俊

Fターム(参考) 5B042 GA11 GA33 GA39 JJ04 JJ23
JJ29 JJ31 KK20

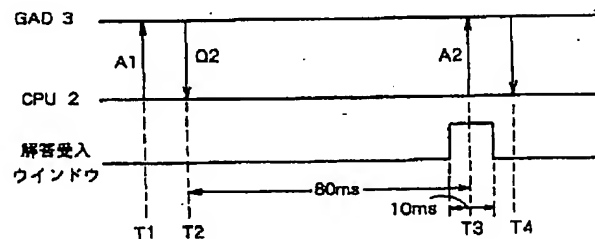
5B045 AA00 GG06 JJ03 JJ04 JJ13
JJ14

(54) 【発明の名称】 複数データ処理装置間相互監視方法

(57) 【要約】

【課題】 データ処理装置間で動作の相互監視を行い、些細な不具合も迅速に検知できるようにすること。

【解決手段】 ランダムに選択された質問番号Q2をCPU2からGAD3へ送り、GAD3では質問番号Q2に対応して定められている演算を行い、その演算結果A2をCPU2に送り出す。CPU2は演算結果A2が解答受入ウィンドウが開いている所定の期間内に送られて来たか否か、及び演算結果A2の内容が正解か否かによりエラーカウンタ31の値を増減し、その結果に応じて制御動作に変更を加える。



【特許請求の範囲】

【請求項 1】 複数のデータ処理装置が正常に動作しているか否かを相互に監視するための方法であって、複数の質問用コードを予め用意しておく、

第 1 のデータ処理装置から第 2 のデータ処理装置に前記複数の質問用コードのうちのいずれか 1 つを送り、前記第 2 のデータ処理装置において受け取った受信質問用コードに対応して予め定められている所定の演算を実行し、

得られた実演算結果を前記第 1 のデータ処理装置に送り返し、

前記第 1 のデータ処理装置において所定の回答期間内に該実演算結果を受け取ることができたか否かの結果及び該実演算結果の内容と前記受信質問用コードに対応して予め用意されている正解演算結果との比較結果に従って前記第 1 及び第 2 のデータ処理装置の動作の監視を行うことを特徴とする複数データ処理装置間相互監視方法が提案される。

【請求項 2】 前記質問用コードが質問番号であり、前記所定の演算がその時受け取った質問番号を演算用データとして含んでいる請求項 1 記載の複数データ処理装置間相互監視方法。

【請求項 3】 前記質問用コードが予め用意されている複数の質問用コードのうちからランダムに 1 つだけ選ばれるようにした請求項 1 記載の複数データ処理装置間相互監視方法。

【請求項 4】 前記実演算結果が前記所定の回答期間内に得られたか否か及び前記実演算結果が正解であったか否かに応じてエラーカウンタの値を増減させ、該エラーカウンタの値が所定の値を越えたときに所定の命令を出力するようにした請求項 1 記載の複数データ処理装置間相互監視方法。

【請求項 5】 前記エラーカウンタの値に応じた命令が出力されるようにした請求項 4 記載の複数データ処理装置間相互監視方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は CPU 或いは CPU に相当するデータ処理機能を備えた機能素子などのデータ処理装置間で相互に監視を行うための、複数データ処理装置間相互監視方法に関する。

【0002】

【従来の技術】 従来から、CPU の動作が予定通りに正常に動作しているか否かをチェックするため、ウォッチドックタイマを用いて CPU におけるプログラムの暴走等の発生を検出する方法が公知であり、この種のウォッチドックタイマを用いて複数の CPU 間で相互監視を行う方法が提案されている。

【0003】 また、特開昭 59-58560 号公報には、主電算機と副電算機との間で演算結果を相互に監視

すると共に、副電算機から主電算機へ演算結果を与えられたときからそれに応ずる演算結果が主電算機から与えられるまでの時間を監視することにより各電算機の異常有無を自動的に診断する方式が開示されている。

【0004】

【発明が解決しようとする課題】 しかし、ウォッチドックタイマを用いる前者の従来方法によると、ウォッチドックパルスが出続けるような不具合が発生した場合にはこれを検出することができないという問題点に加えて、CPU におけるデータ処理において発生した不具合の詳細内容までもを知ることができず、チェック可能なモジュールや機能が少ないという問題点を有している。

【0005】 また、後者の従来技術によると、主電算機においてデータに所定の定数を加算した演算結果を副電算機に送り、副電算機で予め定められた約束にしたがいその判定を行うと共に、そのデータのやりとりにおいてプログラム・ステップ数に応じて予想される応答時間をチェックして異常の発生の有無をチェックする構成であるから、偶然の一致により異常の発生を検出できない場合が生じるほか、その異常内容を細かくチェックすることができない。したがって、例えば車両用の制御装置に使用されているデータ処理装置の如く、車両として不適切な動作を生じさせるような異常処理を迅速に検出し、これに対して適切な処置を施すことが必要とされる用途には不向きである。

【0006】 本発明の目的は、CPU 或いは CPU に相当するデータ処理機能を備えた各種のデータ処理用機能素子などの間で相互に動作チェックを行い、これによりいずれかのデータ処理装置に些細な不具合が発生した場合でもその検知が可能である複数データ処理装置間相互監視方法を提供することにある。

【0007】

【課題を解決するための手段】 上記課題を解決するため、請求項 1 の発明によれば、複数のデータ処理装置が正常に動作しているか否かを相互に監視するための方法であって、複数の質問用コードを予め用意しておく、第 1 のデータ処理装置から第 2 のデータ処理装置に前記複数の質問用コードのうちのいずれか 1 つを送り、前記第 2 のデータ処理装置において受け取った受信質問用コードに対応して予め定められている所定の演算を実行し、得られた実演算結果を前記第 1 のデータ処理装置に送り返し、前記第 1 のデータ処理装置において所定の回答期間内に該実演算結果を受け取ることができたか否かの結果及び該実演算結果の内容と前記受信質問用コードに対応して予め用意されている正解演算結果との比較結果に従って前記第 1 及び第 2 のデータ処理装置の動作の監視を行うことを特徴とする複数データ処理装置間相互監視方法が提案される。

【0008】 請求項 2 の発明によれば、前記質問用コードが質問番号であり、前記所定の演算がその時受け取っ

た質問番号を演算用データとして含んでいる方法が提案される。

【0009】請求項3の発明によれば、前記質問用コードが予め用意されている複数の質問用コードのうちからランダムに1つだけ選ばれるようにした方法が提案される。

【0010】請求項4の発明によれば、前記実演算結果が前記所定の回答期間内に得られたか否か及び前記実演算結果が正解であったか否かに応じてエラーカウンタの値を増減させ、該エラーカウンタの値が所定の値を越えたときに所定の命令を出力するようにした方法が提案される。

【0011】請求項5の発明によれば、前記エラーカウンタの値に応じた命令が出力されるようにした方法が提案される。

【0012】

【発明の実施の形態】以下、図面を参照して本発明の実施の形態の一例につき詳細に説明する。

【0013】図1は、CPUとエーシック（GAD）との間において本発明の方法により両者の動作を相互監視するようにした場合のデータ処理システムの実施の形態の一例を示す概略システム構成図である。

【0014】図1に示したデータ処理システム1は、内燃機関車両に搭載されている車両駆動用内燃機関（図示せず）の制御のためのものであり、助手席の下部等に設置されるコントロールボックス内に設けられている主制御用のCPU2と、図示しない内燃機関へ燃料を供給するための燃料噴射ポンプ（図示せず）の制御のため、コントロールボックスとは別の場所に設置されたGAD3との2つのデータ処理装置を備え、CPU2とGAD3とはアドレス・データバス4によって接続され、これによりCPU2とGAD3との間でデータのやりとりを行うことができる構成となっている。

【0015】GAD3内には、エラーカウンタ31、回答用バッファ32及び質問用バッファ33が、燃料噴射ポンプを制御するための図示しないCPUのメモリ空間内に、以下に詳しく説明するCPU2とGAD3との相互監視のために割り当てられている。

【0016】CPU2とGAD3との間でこれらの動作が予定通り正常に行われているか否かを相互に監視するため、GAD3側には複数の質問内容に対応する質問番号と、これらの質問番号に対応した正解演算結果とが格納されている。本実施の形態では、質問は16種類用意されており、これら16種類の質問には0～Fまでの番号が質問番号として付されている。GAD3においては、これら16種類の質問のうちのいずれか1つの質問番号が乱数発生手段を用いてランダムに選択され、選択された質問番号が質問用バッファ33に書き込まれる構成となっている。

【0017】質問用バッファ33内の質問番号は所定の

タイミングでアドレス・データバス4を介してCPU2に送られ、CPU2では質問番号を受け取ると、この質問番号に対応して予め定められている演算を実行する。

【0018】本実施の形態では、CPU2で実行されるべき演算は、質問番号をnとすると、 $n + 2^5 + 2^7 + 2^9 + (n * 2^{10}) + 2^{14} + 2^{15}$ と定められている。このように演算データとしてnが含まれており、したがって、演算結果は質問番号が異なればそれに応じて異なり、結局、16種類の演算結果が存在することになる。GAD3においては、これら16種類の演算結果が質問番号に対応して図示しないメモリ内に格納されている。そして、CPU2において所要の演算が終了するとその演算結果はアドレス・データバス4を介してGAD3に送られる。

【0019】CPU2に対して質問番号nを送ってからその演算結果がGAD3に戻ってくるまでの時間が所定の一定の値となるようにCPU2のプログラムが設定されており、ここでは、この時間も相互監視のためのチェック項目となっている。本実施の形態では、質問番号nの送出から演算結果受け取りまでの時間Tが80（ms）に設定されており、GAD3では質問番号nの送出から75（ms）～85（ms）までの10（ms）の時間幅の回答受入ウィンドウが設定され、この回答受入ウィンドウが開いている間だけCPU2からの演算結果を受け取ることができるように構成されている。

【0020】回答受入ウィンドウが開いている間に演算結果が送られてきた場合にのみGAD3はこの演算結果を受け取り、回答用バッファ32内に格納する。このように、前回の質問に対する回答が回答用バッファ32内に書き込まれると、GAD3は次の質問番号をランダムに選択し、選択された質問番号を質問用バッファ33にセットする。このセットされた質問番号はCPU2が読み出してその定められた回答時間内に回答することになる。

【0021】回答用バッファ32内に格納された演算結果は、GAD3に予め格納されている正解演算結果と照合され、回答用バッファ32内に格納されている演算結果の正当性がチェックされる。これと同時に、回答受入ウィンドウが開いている間にCPU2から演算結果が送られてきたか否か、すなわちCPU2からの演算結果の送り出しが予定したタイミングで実行されているか否かがチェックされる。

【0022】この2つのチェック結果の適否に応じて、エラーカウンタ31の値が変更される。エラーカウンタ31の値はチェック結果が「否」とする度に増加し、そのカウント値に応じてデータ処理システム1の制御状況が変更されることによりデータ処理システム1の障害状況に応じて内燃機関の運転制御が制限される構成となっている。

【0023】次に、図2を参照しながら、CPU2とG

AD3との間における相互監視のための動作についてより具体的に詳しく説明する。時点T1でCPU2からGAD3へ前回の質問に対する演算結果A1が送出され、GAD3においてその処理が終了すると、時点T2において質問番号Q2がGAD3からCPU2へ送られる。この質問番号Q2は、乱数発生手段を用いて予め用意されている16種類のうちからランダムに選ばれたものである。

【0024】CPU2は、GAD3からの質問番号Q2を受け取ったならば、上述の如く所定の演算を行う。このときGAD3においては時点T2から80(ms)の点に10(ms)幅の回答受入ウィンドウが設定される。データ処理システム1が正常に動作していれば、この回答受入ウィンドウが開かれている期間内に演算結果A2がGAD3に入力されるので、その演算結果A2を回答用バッファ32に格納することができる。若し、データ処理システム1が正常に動作していないと、CPU2からの演算結果A2が回答受入ウィンドウが開いている場合に送られないか、又は送られたとしてもその内容が正解演算結果と一致しないなどの不具合を生じることになる。

【0025】GAD3では、回答受入ウィンドウが開かれている間にCPU2から演算結果A2が送られてきたか否か、及び送られてきたとしてもその演算結果A2の内容が正しいか否かがチェックされ、このチェック結果に応じてエラーカウンタ31のカウントアップ動作が行われる。

【0026】本実施の形態では、エラーカウンタ31の最小値は0であり、回答受入ウィンドウが開かれている間にCPU2から演算結果が送られてき場合にはエラーカウンタ31には-1が加えられる。一方、演算結果が誤っている場合、又は回答受入ウィンドウが開かれている間にCPU2から演算結果が送られてこなかった場合には、エラーカウンタ31に+1が加えられる。エラーカウンタ31の最大値は+7であり、エラーカウンタ31の値が+5に達すると、燃料噴射が停止され、機関の回転が停止することになる。

【0027】このように、エラーカウンタ31のカウント値が増減され、そのカウント値が+5に達したときに、燃料噴射が停止され、機関の回転が停止する構成のほか、エラーカウンタ31のカウント値に応じて内燃機関の制御に、例えば下記の如く、段階的に制限を加える構成とすることもできる。

(1) カウンタ値が4以下の場合は正常とみなし制御動作に制限を加えない。

(2) カウンタ値が4~6の間では、制御動作に対し、機関回転信号の処理を停止することを示すフラグ`flag_disable`をセットし、機関回転信号の処理を停止するという制限を加える。この結果、機関の動作は、気筒判別用に用いるポンプ回転信号`seg`が無効である

と停止する。走行中でポンプ回転信号`seg`が有効であれば変化無し。

(3) カウンタ値が7となると、メインリレーを切断し、コントロールユニットの動作を停止する。この場合には機関の回転が停止する。

【0028】なお、CPU2は、GAD3の監視機能が正常に作動しているか否かをチェックするため、演算結果の内容を不正解なものとするか、或いは回答受入ウィンドウが開いていないタイミングで演算結果をGAD3に送出し、これによりエラーカウンタ31が所定のカウンタアップ動作を行うか否かを確認する動作を適宜のタイミングで実行する構成としてもよい。この場合、上述の確認動作後、エラーカウンタ31のカウント値は確認動作実行前の値に戻される。

【0029】データ処理システム1は以上のように構成されているので、質問番号はランダムに選ばれることになり、その演算結果が偶然に正解となる確率は極めて低く、その監視を確実性をもって行うことができる。また、演算内容をより複雑にすればCPU2での演算のために多数のモジュールを使うことになり、プログラム動作上の些細な不良でも発見することが可能となる。また、演算結果の応答の時間もチェックしているので、監視動作の信頼性は極めて高いものである。

【0030】この結果、CPU2に監視のために必要な演算のための僅かな時間を与えるだけで、CPU2及びGAD3の不良を素早く検知することができ、車両の制御に不適切な状態が生じるのを未然に防止することができる。また、監視のための演算が多少複雑になっても、CPU2の負荷としてはいたしたことがないため、本来の制御動作に悪影響を与えることもない。

【0031】図1に示した実施の形態では、CPU2とGAD3との間で相互に監視を行う場合の例を示したが、GAD3の代わりにCPUを用いたシステム構成であっても本発明を適用できるし、CPU2がGADであるシステム構成であっても本発明を適用することができる。また、ASICを用いたデータ処理装置間又はCPUとASICとの間の相互監視のためにも本発明を同様に適用することができることは勿論である。

【0032】

【発明の効果】本発明によれば、上述の如く、相手方のデータ処理装置に質問コードを送り、この質問コードに対応した演算を実行させてその結果を受け取るようにし、この演算結果の内容とその送出タイミングとに基づいて動作の相互監視を行うようにしたので、偶然性が良好に排除され、その監視動作の信頼性は極めて高いものである。また、演算内容をより複雑にすれば演算のために多数のモジュールを使うことになり、プログラム動作上の些細な不良でも発見することが可能となる。

【図面の簡単な説明】

【図1】本発明の方法によりCPUとGADとの間で動

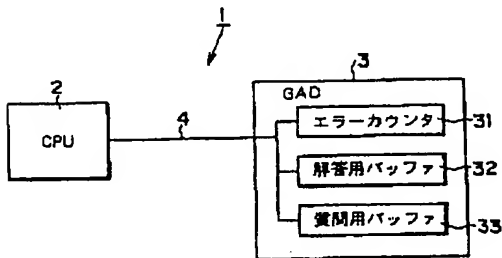
作の相互監視を行うようにしたデータ処理システムの実施の形態の一例を示す概略システム図。

【図2】図1のデータ処理システムにおける相互監視の動作を説明するための図。

【符号の説明】

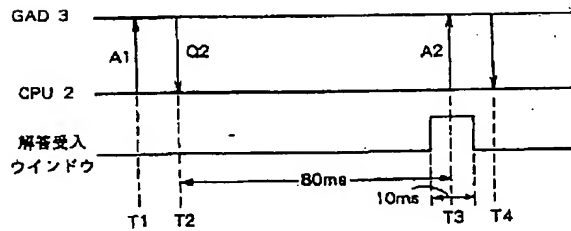
- 1 データ処理システム
- 2 CPU

【図1】



- 3 GAD
- 4 アドレス・データバス
- 31 エラーカウンタ
- 32 解答用バッファ
- 33 質問用バッファ
- A1、A2 演算結果
- Q2 質問番号

【図2】



THIS PAGE BLANK (USPTO)